

On NP-complete problems arising in the control of 2-cast Clos networks

Paweł Obszarski · Andrzej Jastrzębski ·
Marek Kubale

Received: date / Accepted: date

Abstract In this article we study 3-stage Clos networks with 2-cast calls. We investigate various configurations of input and output switches and discuss some routing problems involved in blocking states. To express our results in a formal setting we introduce a model of hypergraph edge-coloring. A new class of bipartite hypergraphs corresponding to Clos networks is studied. We identify some polynomially solvable instances as well as a number of NP-complete cases. This is in sharp contrast to classical unicast Clos networks for which all the control problems are polynomially solvable.

Keywords Clos network · 2-cast calls · hypergraph edge-coloring · 3-uniform hypergraphs · rearrangeable networks · nonblocking networks · NP-completeness

This paper has been partially supported by Narodowe Centrum Nauki under contract DEC-2011/02/A/ST6/00201

P. Obszarski
Department of Algorithms and System Modeling, Gdańsk University of Technology
Tel.: +48 58 347 17 41
Fax: +48 58 347 17 66
E-mail: pawel.obszarski@eti.pg.gda.pl

A. Jastrzębski
Department of Algorithms and System Modeling, Gdańsk University of Technology
Tel.: +48 58 347 22 68
Fax: +48 58 347 17 66
E-mail: jendrek@eti.pg.gda.pl

M. Kubale
Department of Algorithms and System Modeling, Gdańsk University of Technology
Tel.: +48 58 347 18 18
Fax: +48 58 347 17 66
E-mail: kubale@eti.pg.gda.pl

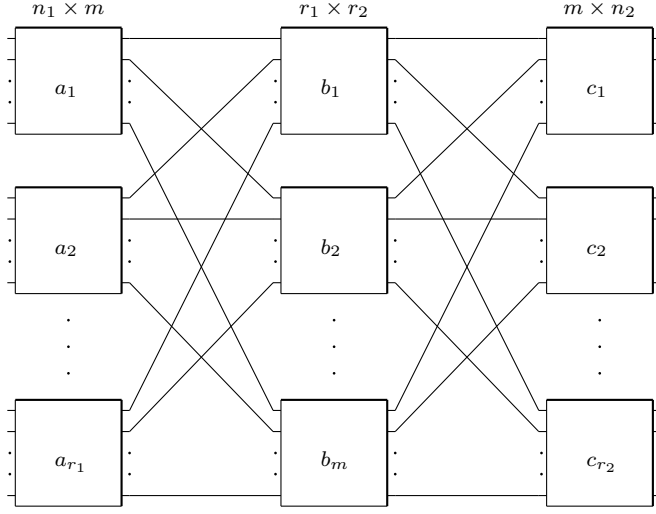


Fig. 1 An example of Clos network $C(n_1, r_1, m, n_2, r_2)$.

1 Introduction

Clos networks were introduced by Charles Clos in his seminal paper [4]. However, our basic notation and terminology follows that of Hwang [5]. In this article we discuss 3-*stage* Clos networks which consist of several switches arranged in 3 layers. Interconnection networks of this type are characterized as follows:

- The first (input) stage consists of r_1 switches (crossbars) each with n_1 inputs and m outputs. We say that such a switch is of size $n_1 \times m$.
- The second (middle) stage consists of m switches each of size $r_1 \times r_2$.
- The third (output) stage consists of r_2 switches each of size $m \times n_2$.
- There exists exactly one link between every middle switch and every input (output) switch.

Clos network with these parameters is denoted $C(n_1, r_1, m, n_2, r_2)$. An example of such a network is presented in Fig. 1.

Clos networks are theoretical idealization of practical multistage switching systems, which means that each connection from an inlet on input switch to an outlet on output switch consists of links that are not to be shared with any other connections.

We recall that in multicast calls each request consists of one idle inlet and a group of idle outlets. A multicast call is said to be *f-cast* if at most f outlets can be requested in one call. Herein we restrict our analysis to 2-cast calls only. To allow multicast connections, the switches with fan-out capability are required. A switch with *fan-out* property is capable of distributing a signal from one input to a few outputs. In our model we assume that only switches

in the second and the third stage have such a property (so-called model 1[6]). Therefore, we assume that $n_1 \leq n_2 \leq m$.

In general, considerations devoted to combinatorial properties of 2-cast Clos networks may be headed into two directions. First, we may try to estimate a sufficient number of middle-stage switches that need to be installed in order to guarantee that the network is rearrangeable and/or nonblocking. This problem is briefly discussed in Section 3. Secondly, we may try to design a routing algorithm that attempts to minimize the number of middle switches used. In this article we mainly point out in Sections 4 and 5 several reasons of intractability involved in the second approach.

2 Edge-coloring of hypergraphs

The problem of routing multicast Clos networks can be modeled by a hypergraph edge-coloring. In particular, 2-cast Clos network can be modeled by an edge-coloring of 3-uniform hypergraph, which is introduced in this section.

Let $H = (V, E)$ be a *hypergraph*, where $V = V(H)$ is a set of *vertices* and $E = E(H)$ is a multiset of nonempty subsets of V called *hyperedges* or simply *edges*. We say that a hyperedge e and a vertex v are *incident* if $v \in e$. Also, two edges that have a vertex in common are said to be *adjacent*. A d -*edge* is a hyperedge that contains exactly d vertices. We say that $d = |e|$ is a *dimension* of hyperedge e . If all edges of a hypergraph are of the same dimension d then the hypergraph is said to be d -*uniform*. A hypergraph is *simple* if each edge is unique (in the sense of vertices it contains). A hypergraph H is *linear* if each pair of edges share at most one vertex. Linearity is stronger than simplicity in the sense that each linear hypergraph is also simple. Notice that simple (linear) 2-uniform hypergraphs are just graphs.

The *degree* $\deg(v)$ of a vertex $v \in V$ is the number of edges in which v occurs. $\Delta(H) = \max_{v \in V} \deg(v)$ is the *degree* of H .

A *proper edge-coloring* (k -*edge-coloring*) of a hypergraph H with k colors is a function $c : E(H) \rightarrow \{1, \dots, k\}$ such that no two adjacent hyperedges get the same color (number). A coloring that uses the minimum number of colors is called *optimal*. The *chromatic index* $\chi'(H)$ of H is defined to be the number of colors in an optimal coloring of H .

For a hypergraph H , the *line graph* $L(H)$ is a simple graph representing adjacency between hyperedges in H . More precisely, each hyperedge of H is assigned a vertex in $L(H)$ and two vertices in $L(H)$ are adjacent if and only if their corresponding hyperedges in H have a vertex in common. It is easy to notice that an edge-coloring of a hypergraph H is equivalent to vertex-coloring of its line graph $L(H)$.

Edge-coloring of hypergraphs can be applied to model the operation of multicast Clos networks in the following manner. Input and output switches correspond to vertices of a hypergraph, connecting paths and connection requests are represented by hyperedges. The middle switch participating in a connection determines the color of the corresponding hyperedge. An example

of reduction from a 2-cast Clos network to edge-coloring of 3-uniform hypergraphs is presented in Fig. 2.

Notice that for a Clos network with $m = 2$ the decision problem whether a new call can be unblocked is easy (in terms of complexity) and is equivalent to 2-edge-coloring of the corresponding hypergraph, which is polynomially solvable.

In this paper a special class of 3-uniform hypergraphs is considered. For this reason we will need some additional notions. In particular, we say that a hypergraph is *bipartite* if its vertex set can be split into two partitions in such a way that each edge has exactly one vertex in the first partition and two vertices in the second one. In connection with Clos networks terminology we will call the first partition the *in-partition* and the second one the *out-partition*. We denote these partitions V_i and V_o , respectively. The maximum degree of a vertex in the in-partition will be called the *in-degree*, in symbols Δ_i , and the maximum degree of a vertex in the out-partition will be called the *out-degree*, in symbols Δ_o .

For a hypergraph $H = (V, E)$ and a set of vertices $U \subseteq V$, we say that hypergraph $H[U]$ is a subhypergraph *induced* by the set U if $H[U] = (U, E')$ where $E' = \{e : e \neq \emptyset \wedge e \in E''\}$ while there exists bijective function $f : E \rightarrow E''$ so that $f(e) = e \cap U$. In other words, each hyperedge of induced hypergraph is an intersection of the set U and certain uniquely assigned hyperedge from E (empty intersections are omitted). For example, $H[V_o]$ is a graph or multigraph since each hyperedge of H has exactly two vertices in V_o . On the other hand, $H[V_i]$ consists of bunches of 1-edges. Such an approach allows us to look at dependences in out- and in-partitions separately.

In the following we mainly discuss 3-uniform hypergraphs. Henceforth, by the term hypergraph we mean a 3-uniform hypergraph, unless otherwise stated. Observe that although we simplify the model by omitting 2-edges (that correspond to 2-cast calls with outlets on the same output switch) we can take them into account, since each 2-edge can be extended to 3-edge by introducing one dummy vertex into it.

3 Upper bounds on the chromatic index

Proper estimation of the number of switches in the central stage is essential at the phase of designing the Clos network structure. In this section we present some relevant conjectures and bounds on that number. We remind that the minimal number of switches in the middle stage cannot be less than $\max\{\chi'(H)\}$, where maximum is among all hypergraphs realized by the network.

We start with a basic bound that applies to all 3-uniform hypergraphs.

Proposition 1 *Each 3-uniform hypergraph H can be edge-colored with $3\Delta(H) - 2$ colors.*

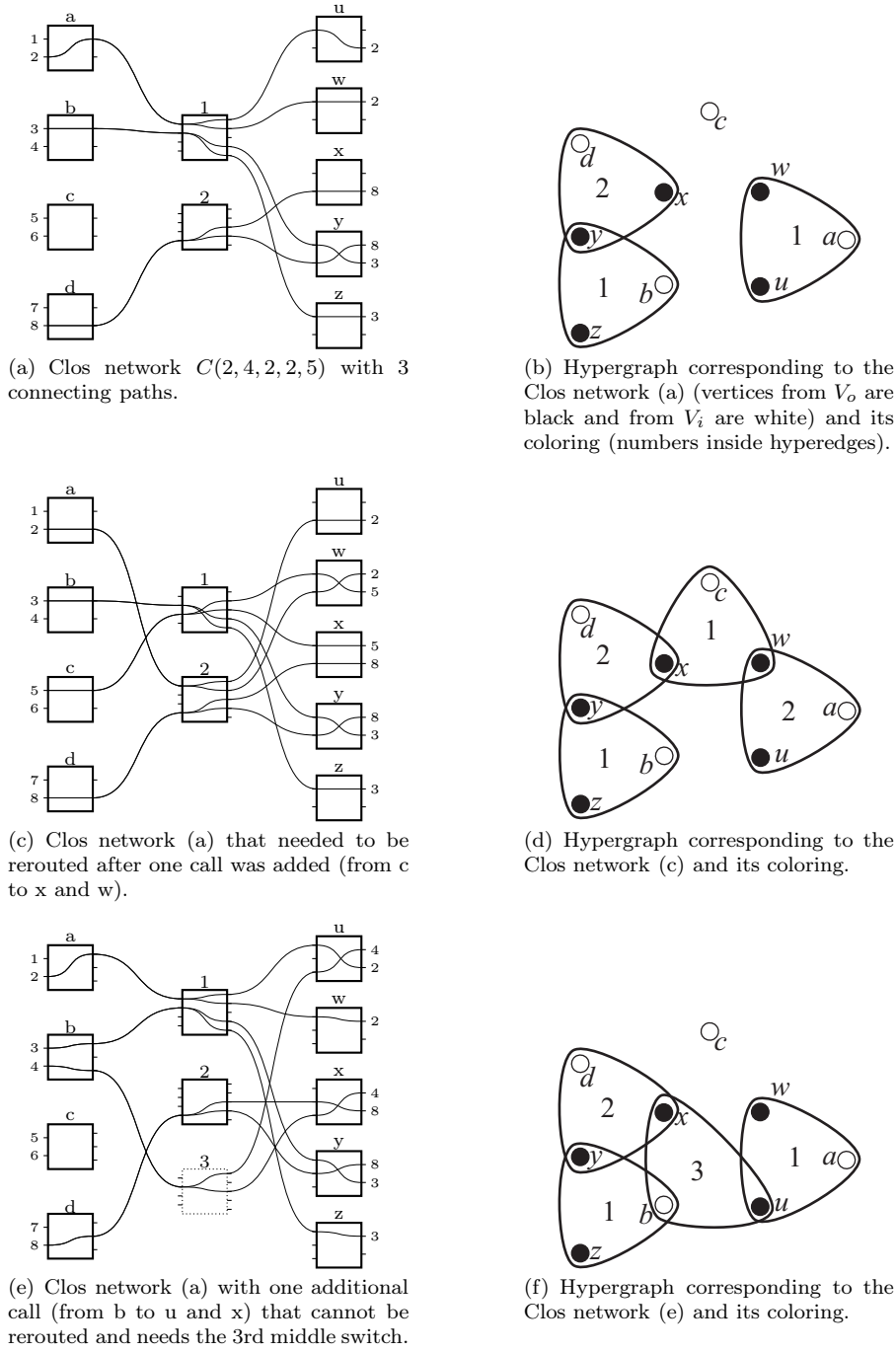


Fig. 2 Example of 2-cast Clos network and corresponding hypergraphs.

To justify this fact it is enough to transform H into its line graph $L(H)$ and apply Brooks inequality to it [1]. This bound guarantees that a new request can never be blocked regardless of the current state of Clos network $C(n_1, r_1, m, n_2, r_2)$ provided that $m \geq 3n_2 - 2$.

There is a much stronger conjecture formulated in [9]. Unfortunately, the problem still remains open.

Conjecture 1 (Hwang, Lin 95 [9]) Every bipartite 3-uniform hypergraph is (2Δ) -edge-colorable.

In terms of Clos network the truth of Conjecture 1 implies that $C(n_1, r_1, m, n_2, r_2)$ is rearrangeable if $m \geq 2n_2$. For some special cases the conjecture has been proven to be true. For example:

- $\Delta_i \geq \Delta_o$ and $\Delta_o \leq 3$ [7]
- $\Delta_i > \Delta_o$ and $|V_o| \leq 4$ [8]
- $|V_i| \leq 4$ [10]

A generalization of Hwang-Lin's conjecture has been proposed in [7].

Conjecture 2 If $\Delta_i \geq \Delta_o$ then bipartite 3-uniform hypergraph is $(\Delta_o + \Delta_i)$ -edge-colorable.

Based on some computational experiments, we propose even stronger conjecture.

Conjecture 3 If $\Delta_i \geq 2\Delta_o$ then bipartite 3-uniform hypergraph is Δ_i -edge-colorable.

If our conjecture is true then so are Conjectures 1 and 2.

4 Complexity of edge-coloring in special cases

In this section we discuss the cases in which $H[V_o]$ forms some highly structured graphs. We give a detailed proof of NP-completeness of deciding the 3-edge-colorability of H when the out-partition is composed of cycles. However, we also consider forests, trees, collections of cliques and collections of multipaths. Finally we indicate a polynomial-time algorithm for the case in which $H[V_o]$ is a collection of stars.

Theorem 1 *It is NP-complete to decide whether a bipartite hypergraph H with $\Delta_i = 3$ has a 3-edge-coloring in the following cases:*

- (a) H is linear and $H[V_o]$ is a collection of cycles C_4 ,
- (b) H is linear, $H[V_o]$ is a forest and $\Delta_o = 3$,
- (c) H is linear, $H[V_o]$ is a caterpillar and $\Delta_o = 3$,
- (d) H is simple and $H[V_o]$ is a collection of complete graphs K_4 ,
- (e) H is simple, $H[V_o]$ is a collection of multipaths on three vertices and $\Delta_o = 3$,

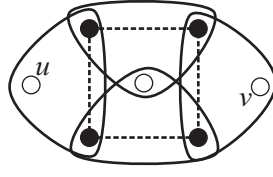


Fig. 3 Gadget for cycle. The dashed lines mark underlying cycle C_4 .

(f) H is simple, $H[V_o]$ is a multipath and $\Delta_o = 3$.

Proof Obviously, the problem is in NP, since checking whether coloring is proper requires only verifying if adjacent edges are of different color. Below we consider 6 cases. In each case the proof is based on the fact that 3-edge-coloring of a cubic graph is an NP-complete problem [3].

Case (a)

Let us consider a cubic graph G . We transform G to a 3-uniform and bipartite hypergraph H by replacing each edge $\{u, v\} \in E(G)$ with a gadget shown in Fig. 3. Dashed line marks a graph C_4 in $H[V_o]$. The gadget is joint to the graph via vertices u and v . White vertices constitute the in-partition. Note, that $\Delta_i = 3$ as the initial graph is cubic. The black vertices are part of the out-partition. Together with the hyperedges of gadget they form cycles C_4 , one for each gadget. Finally, we have to explain why graph G is 3-colorable if and only if edges of the corresponding hypergraph H can be colored with three colors. First, let us assume that we know a 3-edge-coloring of graph G . Then, consider each gadget separately, and color the hypergraph H in the following manner.

- the far left hyperedge and the far right hyperedge get the same color as that of the edge $\{u, v\}$ of G replaced by the gadget.
- two hyperedges inside the gadget get the remaining two colors.

One can see that each gadget of hypergraph is properly colored with three colors. Hence the whole hypergraph can also be colored with three colors.

Now let us assume that we have a 3-edge-coloring of the hypergraph H . The crucial point is that hyperedges on the both sides of a gadget have to be colored with the same color in each proper 3-edge-coloring. Otherwise, since the inner side hyperedges are adjacent to both outer side hyperedges and to each other, the fourth color would have to be used. On the basis of this fact, all we need to do is to color the edges of graph G with the colors of gadgets' outer side hyperedges.

Case (b)

Given any cubic graph G , replace each edge $\{u, v\}$ of it with the gadget presented in Fig. 4. The dashed lines mark a tree in $H[V_o]$. We connect the gadget via vertices u and v . Bearing in mind the proof of case (a), to make this one valid we need to show that the gadget can be edge-colored with three colors and each proper 3-coloring requires edges e_1 and e_2 to be of the same color.

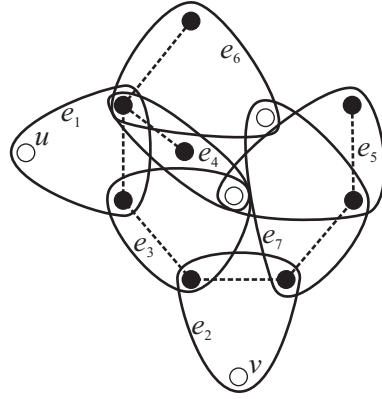


Fig. 4 Gadget for forest. The dashed lines mark an underlying tree.

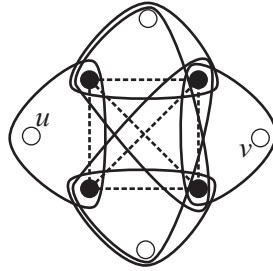


Fig. 5 Gadget for complete graphs. The dashed lines mark underlying K_4 .

Edges e_1 , e_3 and e_4 are adjacent to each other, so they should be colored with three distinct colors. Let us assume that these colors are c_1 , c_2 and c_3 , respectively. Observe that e_5 needs to be colored with the same color as e_1 in each 3-coloring, so it gets c_1 . This and the fact that e_6 is colored with c_2 , implies that e_7 is colored with c_3 . So edge e_2 in each proper 3-edge-coloring must be colored with c_1 .

Case (c)

We have proven that the problem is hard for a set of trees (forest). It occurs that it is enough to connect gadgets together via a vertex of degree 1 in edges e_6 or e_5 using some additional edges in order to get a caterpillar.

Case (d)

The proof remains very similar to that for cycles C_4 . Obviously, the gadget is different (see Fig. 5) but argument remains the same. Hence, we leave detailed considerations to the reader.

Case (e)

Yet another proof similar to that of case (a). The gadget that replaces edges of G is presented in Fig. 6.

Case (f)



Fig. 6 Gadget for multipath. The dashed lines mark an underlying multipath.

Notice that if we connect gadgets from Fig. 6 together via vertices of degree 1 and some additional hyperedges we get single multipath in $H[V_o]$.

It has been shown in the proof of Theorem 1(a) that in considered hypergraphs $\Delta_i = 3$. Also note that $\Delta_o = 2$, since the $H[V_o]$ is a collection of cycles in that case. Hence we have:

Corollary 1 *Edge-coloring of bipartite hypergraphs is NP-complete even for hypergraphs with $\Delta_i = 3$ and $\Delta_o = 2$.*

In practice this means that there exists no polynomial-time algorithm for rearranging general multicast Clos networks unless $P = NP$. This is in sharp contrast to unicast Clos networks for which we have rearranging algorithms that run in polynomial time [11,12].

However, the problem becomes polynomial if both the in-degree and the out-degree are bounded by 2. Then we have the degree configuration $\Delta_i = \Delta_o = 2$, which will be claimed to be polynomial in Section 5. The edge-coloring problem also becomes polynomial if we decrease the size of cycles and consider a collection of triangles C_3 in the out-partition (for any in-degree). Such case is equivalent to the situation described in Theorem 2.

Let us slightly simplify the problem from Theorem 1(e) to the one with $H[V_o]$ being a collection of multipaths P_3 . It is not difficult to see that the problem becomes polynomial. First observe, that the dependences in terms of edge-coloring are the same for multipaths on three vertices, multicycles based on C_3 and stars. Simply each edge is then adjacent to each other. In the following we look closer to the cases with $H[V_o]$ composed of a number of stars.

Theorem 2 *There exists an $O(|E(H)| \log \Delta(H))$ -time algorithm for optimal edge-coloring of bipartite hypergraph H in which $H[V_o]$ is a collection of stars.*

Proof Consider a bipartite hypergraph H in which a set of stars constitute the out-partition. Notice that in such a configuration each hyperedge has at least one vertex of degree 1. Furthermore, such a vertex must be in out-partition and is a leaf of a star in $H[V_o]$. Since such vertices do not influence edge-coloring so we get rid of them and leave 2-edges only (see Fig. 7). Obviously, the remaining graph is a bipartite multigraph. Clearly, the edge-coloring of H



Fig. 7 Illustration for the proof of Theorem 2. (a) Example of a bipartite hypergraph. (b) Bipartite multigraph corresponding to (a).

is equivalent to edge-coloring of such a multigraph, which can be obtained in time $O(|E| \log \Delta)$ [2].

5 Complexity status for bounded degree hypergraphs

In relation with Clos networks the restriction on Δ_i may stand for the maximal number of calls from a single switch of the first stage. Similarly Δ_o restricts the number of calls leaving any switch form the third stage. We may write that considered networks are of type $C(\Delta_i, r_1, m, \Delta_o, r_2)$.

Table 1 collects the information about the complexity status for various combinations of Δ_i and Δ_o . Below we explain the crucial points of this table.

Table 1 The complexity of edge-coloring depending on the in- and out-degrees.

	$\Delta_i = 1$	$\Delta_i = 2$	$\Delta_i = 3$	$\Delta_i = d_i$
$\Delta_o = 1$	linear	linear	linear	linear
$\Delta_o = 2$	linear	linear [13]	NP-hard	NP-hard
$\Delta_o = 3$	NP-hard	NP-hard	NP-hard	NP-hard
$\Delta_o = d_o$	NP-hard	NP-hard	NP-hard	NP-hard

For cases with $\Delta_o = 1$, the $H[V_o]$ is a simple matching. Then the model describes bunches of hyperedges sharing at most one vertex from V_i . In this case the edge-coloring problem is trivial.

If $\Delta_i = 1$, the in-partition consists of r_1 vertices, each of which is either isolated or incident with one hyperedge. As noticed in the proof of Theorem 2, such vertices do not influence the edge-coloring. This implies that this is the structure of $H[V_o]$ that determines the hypergraph colorability. As long as the edge-coloring problem is linear for graphs with $\Delta = 2$ and NP-complete if $\Delta \geq 3$ [3], the edge-coloring of bipartite hypergraphs is linear for $\Delta_i = 1$ and $\Delta_o \leq 2$, and it is NP-complete if $\Delta_i = 1$ and $\Delta_o \geq 3$.

If $\Delta_i = 2$ and $\Delta_o = 2$, then the line graph of such a bipartite hypergraph is of degree at most 3, hence its vertex-coloring is linear [13]. In terms of Clos networks this means that there exists an efficient algorithm for rearranging 2-cast networks $C(2, r_1, 3, 2, r_2)$, unless $L(H)$ contains K_4 . An example of the

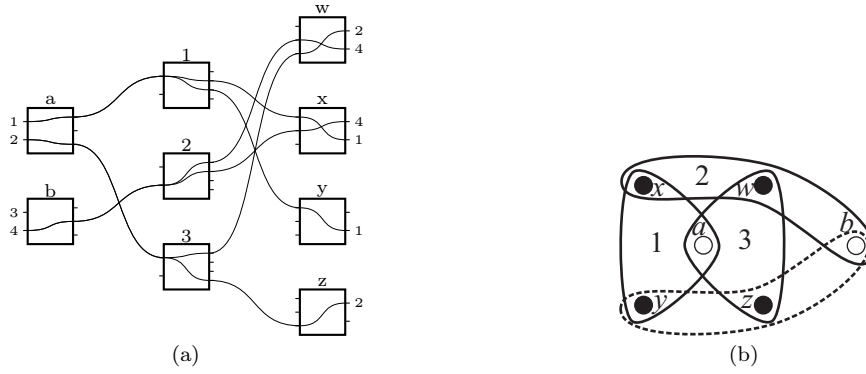


Fig. 8 (a) Clos network $C(2, r_1, 3, 2, r_2)$ with 3 connecting paths. A new call (from b to y and z) would be blocked; (b) Corresponding hypergraph. The new call is marked with dashed line.

smallest network with such a blocking state is shown in Fig. 8. Note that 2-cast networks $C(2, r_1, 4, 2, r_2)$ are nonblocking.

The fact that the problem is NP-complete for $\Delta_o = 2$ and $\Delta_i = 3$ is noted in Corollary 1.

6 Open problems

We conclude with some open questions concerning the colorability of 3-uniform hypergraphs, which in our opinion are well-worth studying. What is the complexity of optimal coloring if $H[V_o]$ is:

- a simple path?
- a simple cycle?
- a simple complete graph?
- a simple complete bipartite graph?
- a collection of cycles C_5 ?
- a forest and $\Delta_i = 2$?

References

1. Brooks R. L.: On colouring the nodes of a network, *Proc. Cambridge Philosophical Society, Math. Phys. Sci.*, vol. 37, pp. 194-197 (1941).
2. Cole R., Ost K., Schirra S.: Edge-coloring bipartite multigraphs in $O(E \log D)$ time, *Combinatorica*, vol. 21, pp. 5-12 (2001).
3. Holyer I.: The NP-completeness of edge-colouring, *SIAM J. Comput.*, vol. 10, pp. 718-720 (1981).
4. Clos C.: A study of nonblocking switching networks, *Bell Syst. Tech. J.*, vol. 32, pp. 406-424 (1953).
5. Hwang F.K.: *The Mathematical Theory of Nonblocking Switching Networks*, World Scientific (2004).

6. Hwang F.K.: A Unifying Approach to Determine the Necessary and Sufficient Conditions for Nonblocking Multicast 3-Stage Clos Networks, *IEEE Trans. on Commun.*, vol. 53, pp. 1581-1586 (2005).
7. Du D.Z., Ngo H.Q.: An extension of DHH-Erdos conjecture on cycle-plus-triangle graphs, *Taiwan J. Math.*, vol. 6, pp. 65-105 (2002).
8. Hwang F.K., Liaw S.-C., Tong L.D.: Strictly nonblocking 3-stage Clos networks with some rearrangeable multicast capability, *IEEE Trans. Commun.*, vol. 6, pp. 261-267 (2002).
9. Hwang F.K., Lin C.H.: Broadcasting in a three-stage point-to-point nonblocking network, *Inter. J. Rel. Qual. Safety Eng.*, vol. 2, pp. 299-307 (1995).
10. Fu H.-L., Hwang F.K.: On 3-Stage Clos Networks with different nonblocking requirements on two types of calls, *J. of Comb. Opt.*, vol. 9, pp. 263-266 (2005).
11. Kubale M.: Average and worst-case performance of Paull's algorithms for rearranging three-stage connection networks, *Annales Des Telecommunications*, vol. 40, pp. 270-276 (1985).
12. Paull M. C.: Reswitching of connection networks. *Bell Syst. Tech. J.*, vol. 41, pp. 833-855 (1962).
13. Skulrattanakulchai S.: Δ -list vertex coloring in linear time, *LNCS*, vol. 2368, pp. 240-248 (2002).